

## DISEÑO E IMPLEMENTACIÓN DE FILTROS DIGITALES PARA AUDIO CON EL PROCESADOR TMS320C6711

Luis Gabriel Marín Collazos

Artículo recibido el 28 de Febrero de 2010, aprobado para publicación el 28 de Mayo de 2010.

### Resumen

Actualmente la implementación hardware de aplicaciones que incrementan las especificaciones en términos de tiempo de respuesta, consumo de potencia y portabilidad se han convertido en una necesidad. Los filtros digitales hacen parte de las aplicaciones costosas computacionalmente y su amplia gama de aplicaciones hacen que sea de interés su implementación en dispositivos empujados. Por otro lado, el procesamiento de audio cubre varios campos de aplicación. Entre los más representativos se pueden mencionar: los sistemas de reconocimiento de patrones de voz humana, reproductores digitales de audio, los efectos de audio que son utilizados en la grabación musical y el procesamiento de la voz empleada en las telecomunicaciones, por nombrar sólo algunos. Este artículo presenta el diseño e implementación de filtros digitales para audio en tiempo real sobre un procesador de señales digitales, mostrando una solución a este tipo de aplicaciones. El procesamiento digital de las señales se realiza mediante filtros de respuesta al impulso infinita y respuesta al impulso finita. Los resultados muestran que un oído humano no entrenado percibe las modificaciones frecuenciales de las señales en el rango audible. Además, se muestra que es posible obtener tiempos de respuesta para esta clase de filtros digitales en el orden de los microsegundos sobre ésta plataforma hardware.

**Palabras clave:** Sistemas empujados, Procesadores de señales digitales, Sistemas Lineales e invariantes en el tiempo, Tiempo Real, Señal Discreta.

## DESIGN AND IMPLEMENTATION OF DIGITAL FILTERS FOR AUDIO WITH THE PROCESSOR TMS320C6711

### Abstract

Currently, the hardware implementations of applications that increase the specifications in terms of response time, power consumption and portability have become a necessity. Digital filters are part of the computationally expensive applications and their wide range of applications make it of interest its implementation in embedded devices. On the other hand, the audio processing covers several fields of application. Among the most representative may include: the pattern-recognition systems of the human voice, digital audio players, audio effects that are used in musical recording and voice processing used in telecommunications, to name a few. This paper presents the design and implementation of digital filters for real time audio on a digital signal processor, showing a solution to this type of application. The digital signal processing is performed by infinite impulse response and finite impulse response filters. The results show that an untrained human ear perceives frequency modifications of the signals in the audible range. Furthermore, it is shown that is possible to obtain response times for this class of digital filters in the order of microseconds on this hardware platform.

**Key words:** Embedded Systems, Digital Signal Processors, Linear systems and time-invariant, Real Time, Discrete Signal.

## Introducción

El tratamiento de señales se relaciona con un conjunto de disciplinas entre las que se encuentran las telecomunicaciones, el control, la industria militar, procesamiento de audio y video, la exploración del espacio, la medicina y la arqueología, por nombrar sólo unas pocas (Oppenheim y Schafer, 2000; Sidney et al. 1998).

Antes de los años 60, la tecnología para el tratamiento de señales era casi exclusivamente tecnología analógica, pero la rápida evolución de los computadores y microprocesadores digitales, junto con algunos importantes desarrollos teóricos como el algoritmo de la transformada rápida de Fourier (FFT), fueron la causa de un importante desplazamiento hacia las tecnologías digitales, naciendo así el campo de tratamiento digital de señales (Barrero et al. 2005).

El tratamiento digital de señales trata de la manipulación matemática de información adquirida sobre un fenómeno físico. Las operaciones involucradas en estos procesos son computacionalmente costosas y para su ejecución en tiempo real, son necesarios dispositivos específicos que cumplan con los requerimientos de la aplicación (Ackenhusen, 1999; Oppenheim y Sillsky, 1998).

Un desarrollo importante en la historia del procesamiento de señales ocurrió en el terreno de la microelectrónica. Los primeros microprocesadores eran lentos para implementar en tiempo real la mayoría de los sistemas en tiempo discreto (Proakis y Manolakis, 1998). A mediados de los ochenta la tecnología de los circuitos integrados avanzó hasta el nivel de permitir la realización de procesadores de propósito específico con características específicas a nivel estructural, que permiten maximizar su rendimiento en términos de capacidades de memoria y velocidad de procesamiento. A estos procesadores se les conoce por el acrónimo de DSP (Digital Signal Processor). Con esta tecnología llegó, por primera vez, la posibilidad de implementar las técnicas de tratamiento de

señales en tiempo discreto con bajo costo, tamaño miniaturizado y bajo consumo de potencia (Wanhammar, 1999; Smith, 1997).

El conocimiento de los DSP en el desarrollo del procesamiento digital de señales ofrece una ventaja en comparación con el tratamiento analógico. Esto debido a la flexibilidad que presenta el software en las modificaciones de sus funciones. Estos sistemas se convierten en más fiables y económicos en donde la única limitación existente queda en la sagacidad del diseñador (Chassaing, 2002; Embree, 1995; Gottfried, 1998).

Hoy en día, el dominio de aplicaciones relacionadas con el procesamiento digital de señales ha crecido. Se pueden mencionar algunos trabajos que realizan procesamiento digitales de señales de audio. Por ejemplo, Grijota (2008), utiliza técnicas de procesado digital de señales de audio en tiempo real, obteniendo algunos de los efectos más frecuentes con los que se puede modificar el sonido original de un instrumento musical, proporcionando al músico un potente vehículo para la expansión de sus capacidades expresivas. Estos efectos fueron implementados en el DSP TMS320c6713 de Texas Instruments.

Pérez (2006), implementó un procesador digital de audio de calidad profesional, basado en un DSP, orientado a darle efectos a la señal de una guitarra eléctrica. Este trabajo presenta diferentes posibilidades de procesado digital de señales de audio para guitarra, como una solución versátil y potente al problema creativo de la obtención de efectos, así como proporcionar a dicho campo, un rigor ingenieril suficiente.

Finalmente, Landa *et al.* (2007), intentan igualar o mejorar la calidad de audio de CD a través de fuentes analógicas como un LP, cassette o instrumentos musicales. Para el desarrollo del proyecto se usa un DSP para hacer un procesamiento en tiempo real y poder crear un dispositivo que convierta una señal analógica con ruido en una señal de audio digital de alta fidelidad, evitando la pérdida de información

en la reproducción final.

Este trabajo presenta el diseño e implementación de filtros digitales en plataformas hardware como una solución cuando se requieran aplicaciones en tiempo real, bajo consumo de potencia y tamaño miniaturizado.

Este artículo está organizado de la siguiente manera: la sección II presentan los materiales y métodos, en la sección III presentan los resultados, discusiones y se exponen las conclusiones.

### **Materiales y métodos**

Para la implementación de los efectos de audio propuestos, se utilizó el procesador DSP TMS320C6711 de Texas Instruments, que está sobre la tarjeta DSK C6711, debido a las ventajas que brinda en el procesamiento digital de señales. A continuación se describe el sistema utilizado para el desarrollo y la metodología empleada.

### **Sistema de Desarrollo DSP**

Los filtros digitales para audio se implementan en el kit de inicio de desarrollo (DSK) C6711. El DSK está compuesto básicamente por el procesador de punto flotante TMS320C6711, dos relojes, memoria SDRAM externa de 16 Mbytes, memoria Flash externa de 128 Kbytes, un ADC de 16 bits con una frecuencia de muestreo de 8 KHz, un controlador JTAG, que provee una fácil emulación y depuración (Texas Instruments, 2000).

El procesador tiene seis unidades aritmético-lógicas (ALU's) y dos multiplicadores. Cuenta con ocho bits de guarda para evitar el overflow aritmético y 32 registros de 32 bits de propósito general. El ciclo máquina de este procesador es de 6,7 ns y es capaz de ejecutar 8 instrucciones en un solo ciclo de reloj. Puede manejar buses de datos de 8, 16 y 32 bits y buses de direcciones de 32 bits. Tiene una arquitectura de memoria interna basada en L1 y L2. La memoria de caché L1 se divide en dos bloques: 4 Kbyte son para memoria de programa y 4 Kbyte para

memoria de datos. La memoria L2 tiene una capacidad de 64 Kbyte, unificada para programa y datos (Texas Instruments, 2000).

La memoria externa SDRAM se puede manejar como un banco de memoria de 4M para palabras de 32 bits con un tiempo de acceso de 10 ns. En cada aplicación se debe configurar la memoria para indicar como queda distribuida físicamente. Las secciones para memoria de programa, memoria de datos inicializados o constantes y datos no inicializados, se deben definir (Texas Instruments, 2000).

### **Metodología**

Los filtros digitales implementados se realizan partiendo de un algoritmo para la adquisición y salida de datos desde el DSP. Seguidamente, se realiza el diseño de cada uno de los filtros digitales utilizando las funciones de transferencia que permiten la realización de los efectos de audio. Estos filtros digitales son implementados en el paquete computacional Matlab®, con el fin de tener una simulación de lo que se desea obtener en el procesador.

Una vez validados los filtros digitales se procedió a escribir el código C apropiado para la plataforma hardware. El código C construido es compilado, ensamblado, depurado y cargado al DSP utilizando el software de evaluación específico Code Composer Studio (CCS). Una comunicación DSP-PC se utilizó por medio del puerto paralelo y el controlador JTAG para poder obtener los datos utilizados en el proceso de validación. Posteriormente, se calculó el Error Cuadrático Medio (ECM) y el Error Absoluto Medio (EAM). También se registran los tiempos promedios consumidos en el procesamiento de las muestras de audio ingresadas a los filtros digitales.

A continuación se realiza la descripción del diseño de los filtros y finalmente se presenta el procedimiento y pruebas realizado para la validación de los filtros digitales para audio.

### **Diseño de los filtros**

La reverberación y la ecualización son los

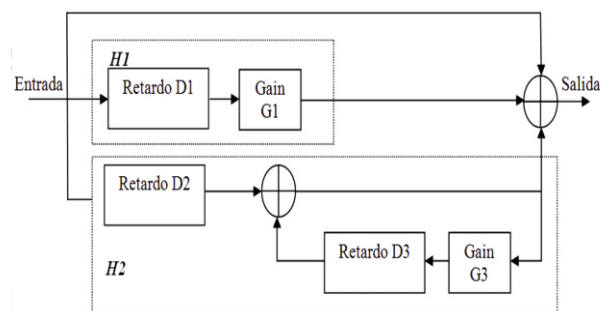
efectos de audio seleccionados para la implementación, dado que éstos se pueden apreciar claramente y se utilizan en el ambiente artístico musical. De igual manera, son efectos usados en los equipos de audio comerciales.

Como el objetivo es procesar señales de audio, lo esencial es conocer el rango de audición humana, que generalmente está considerado entre 20 Hz a 20 KHz, pero este es mucho más sensible para sonidos entre 1 KHz y 4 KHz. Por lo tanto, se seleccionó un espectro de 4 KHz para el diseño de la aplicación.

### Reverberación

La reverberación es el resultado de varias reflexiones del sonido que ocurren dentro de una habitación. La onda de sonido reflejada llegará al oído con cierto retardo respecto a la onda directa, pues ésta viaja una mayor distancia. Además es generalmente más débil dado que las paredes y otras superficies de la habitación absorben parte de la energía del sonido. Por supuesto que la onda reflejada puede rebotar en varias paredes antes de llegar a los oídos. Esta serie de atenuaciones y retardos es lo que se llama reverberación y es el efecto que genera sensación de espacio de una habitación (Grijota, 2008).

Con el diagrama de bloques mostrado en la Figura 1, se obtienen reverberaciones de diferentes lugares, con una buena aproximación.



**Figura 1.** Diagrama de bloques para implementar la reverberación.

Como se puede observar el bloque demarcado como H1, tiene su función de transferencia en el dominio Z, tal como se muestra en la ecuación

(1):

$$H_1(Z) = G1 \cdot Z^{-D1} \quad (1)$$

Esta función de transferencia corresponde a un filtro FIR causal. El bloque demarcado como H2, tiene una función de transferencia en el dominio Z, como se muestra en la ecuación (2):

$$H_2(Z) = \frac{Z^{-D2}}{1 - G3 \cdot Z^{-D3}} \quad (2)$$

En este caso el filtro es un IIR causal. La salida total del sistema para la reverberación se obtiene sumando la entrada más los filtros representados por H1 y H2.

Con la selección apropiada de los parámetros de la Figura 1, se puede fingir la respuesta de cualquier sala, como el de un cuarto de estudio que suele ser una sala con mucho aislamiento sobre las paredes, a lugares sumamente vivaces como un estadio o iglesia (Grijota, 2008).

De los datos sobre diversos espacios acústicos, se deben tener dos parámetros disponibles, el tiempo inicial de demora (TID) y tiempo de reverberación. El TID es el tiempo entre la generación de un sonido y primer eco. El tiempo de reverberación es el tiempo en el que un sonido decrece su amplitud a 1/1000 de su amplitud inicial.

En el Cuadro 1, se indica el TID y el tiempo de reverberación, para diferentes lugares, con estos datos se diseñan los filtros que se implementan en el DSP.

**Cuadro 1.** TID y tiempo de reverberación para diferentes lugares.

Lugar	TID ( ms)	Tiempo de reverberación (s)
Estudio	20	0,40
Auditorio	40	0,75
Sala de concierto	25	1,80
Iglesia	50	3,25

Para la implementación de la reverberación se definen las funciones de transferencia de los bloques H1 y H2 de la Figura 1. Estas funciones de transferencia se presentan en la Cuadro 2.

La salida total del sistema se obtuvo sumando

**Cuadro 2.** Funciones de transferencia para diferentes lugares.

Lugar	H1	H2
Estudio	$H_1(Z) = 0,42 \cdot Z^{-160}$	$H_2(Z) = \frac{Z^{-320}}{1 - 0,42 \cdot Z^{-400}}$
Auditorio	$H_1(Z) = 0,63 \cdot Z^{-320}$	$H_2(Z) = \frac{Z^{-640}}{1 - 0,63 \cdot Z^{-400}}$
Sala de concierto	$H_1(Z) = 0,83 \cdot Z^{-200}$	$H_2(Z) = \frac{Z^{-400}}{1 - 0,83 \cdot Z^{-400}}$
Iglesia	$H_1(Z) = 0,89 \cdot Z^{-400}$	$H_2(Z) = \frac{Z^{-800}}{1 - 0,89 \cdot Z^{-400}}$

la señal de entrada y las señales procesadas por H1 y H2.

### Ecualizador

Un ecualizador permite alterar los niveles de una señal en ciertas frecuencias o bandas de frecuencias, dentro del ancho de banda de dicha señal. Para la implementación de un ecualizador se debe decidir con que filtros digitales se va a realizar, pueden ser filtros FIR e IIR (Marín, 2002).

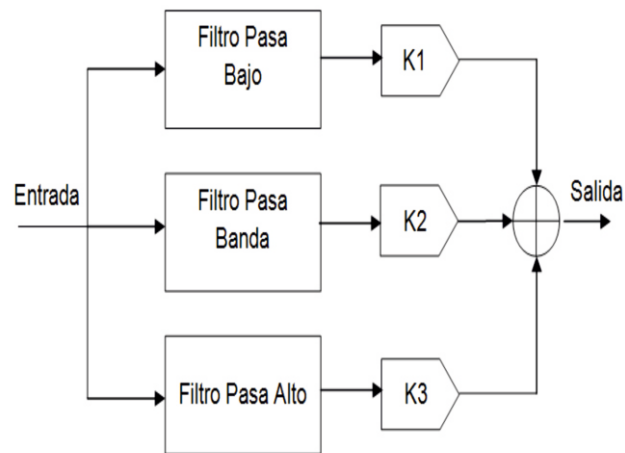
En el desarrollo del ecualizador, se utilizaron las dos topologías anteriormente mencionadas. En la implementación mediante filtros FIR se utilizó un ecualizador de tres bandas y con la utilización de filtros IIR se realizó un ecualizador gráfico.

### Ecualizador de tres bandas

La estructura del ecualizador implementado es mostrada en la Figura 2. Se pueden ver tres filtros FIR que se ejecutan en paralelo y son modificados por una ganancia asociada. La salida total del sistema se obtiene de la suma de los tres filtros.

El diseño del sistema mostrado en la Figura 2, se realizó haciendo uso del paquete computacional Matlab®, con el método de enventanado Hamming y orden 30.

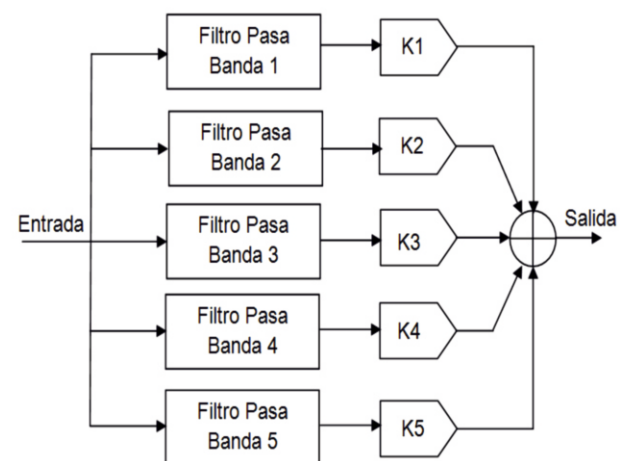
El filtro pasa bajo se diseño para una frecuencia

**Figura 2.** Ecualizador de tres bandas.

de corte en 500 Hz, el filtro pasa banda para una frecuencia central en 2000 Hz con un ancho de banda de 1000 Hz y el filtro pasa alto para una frecuencia de corte en 3000 Hz. La distribución de estas frecuencias se hizo de tal manera que se cubriera todo el rango posible de la voz humana.

### Ecualizador Gráfico

En el caso de un ecualizador gráfico se utilizan filtros pasa-banda de banda angosta. Las bandas normalmente se distribuyen en fracciones de octava. Una octava por encima de una frecuencia significa el doble de esa frecuencia.

**Figura 3.** Estructura de un ecualizador con filtro IIR pasa – banda



La estructura del ecualizador implementado, es mostrada en la Figura 3, en la cual se pueden ver cinco filtros IIR que se ejecutan en paralelo, para obtener la salida total del sistema. Para modificar la curva de respuesta total hace falta cambiar solo la ganancia de cada filtro.

La ecuación que describe la función de transferencia total del ecualizador gráfico, se presenta en la ecuación (3):

$$H[z] = \sum_{i=1}^N G_i H_i(z) \quad (3)$$

Donde  $H_i(z)$  es un filtro IIR de orden dos, tal como se muestra en la ecuación (4):

$$H_i(Z) = \frac{1 - Z^{-2}}{1 + a_1 \cdot Z^{-1} + a_2 \cdot Z^{-2}} \quad (4)$$

Para estos filtros los valores de  $a_1$  y  $a_2$ , se representa en la ecuación (5):

$$\begin{aligned} a_1 &= -2 \cdot R \cdot \cos(w_0) \\ a_2 &= R^2 \end{aligned} \quad (5)$$

Donde:

$R = 1 - BW/2$ .

$w_0$ : Frecuencia central normalizada.

BW: Ancho de banda normalizado.

Para los filtros pasa banda se utilizó un ancho de banda de 80 Hz y la frecuencia central de cada filtro se encuentra al doble de la anterior.

**Cuadro 3.** Funciones de transferencia de los cinco filtros del ecualizador gráfico.

FRECUENCIA	FUNCIÓN DE TRANSFERENCIA
220 Hz	$H(Z) = \frac{1 - Z^{-2}}{1 - 1,90 \cdot Z^{-1} + 0,94 \cdot Z^{-2}}$
440 Hz	$H(Z) = \frac{1 - Z^{-2}}{1 - 1,82 \cdot Z^{-1} + 0,94 \cdot Z^{-2}}$
880 Hz	$H(Z) = \frac{1 - Z^{-2}}{1 - 1,49 \cdot Z^{-1} + 0,94 \cdot Z^{-2}}$
1760 Hz	$H(Z) = \frac{1 - Z^{-2}}{1 - 0,36 \cdot Z^{-1} + 0,94 \cdot Z^{-2}}$
3520 Hz	$H(Z) = \frac{1 - Z^{-2}}{1 - 1,80 \cdot Z^{-1} + 0,94 \cdot Z^{-2}}$

Las frecuencias centrales se localizan en 220 Hz, 440 Hz, 880 Hz, 1760 Hz y 3520 Hz, por lo tanto se calcularon los coeficientes  $a_i$  y  $b_i$  de cada uno de los cinco filtros, quedando de la forma mostrada en el Cuadro 3.

Una vez establecidas las funciones de transferencia, se aplicaron de forma paralela a la señal de entrada, con esto se obtuvo la salida total del sistema. Cada filtro tiene una ganancia controlada.

### Procedimiento y pruebas

El procedimiento utilizado para validar los resultados de salida de los filtros digitales para audio, se describe a continuación:

- 1 Se generan  $N=1000$  muestras provenientes de una señal de audio muestreada a 8 KHz como entrada al filtro digital.
- 2 El valor de salida obtenido de la simulación de los filtros digitales realizada mediante Matlab® se guarda como  $out_{ref}$  y el valor de salida del mismo filtro digital obtenido de la plataforma DSP se guarda como  $out_{DSP}$ .
- 3 Se calcula el ECM y EAM como sigue:
 
$$ECM = \frac{1}{N} \cdot \sum_{i=1}^N (out_{ref}(i) - out_{DSP}(i))^2 \quad (6)$$

$$EAM = \frac{1}{N} \cdot \sum_{i=1}^N abs(out_{ref}(i) - out_{DSP}(i)) \quad (7)$$

Este procedimiento se repite para los seis filtros digitales implementados.

Para el cronometraje de los tiempos consumidos en el procesamiento de cada una de las muestras de audio ingresadas a los filtros digitales, se emplea el siguiente procedimiento:

- 1 Se genera una muestra de audio aleatoria como entrada al filtro digital.
- 2 Se registran los tiempos consumidos en el procesamiento de dicha muestra.
- 3 Se repite 1 y 2 para 50 muestras de audio aleatorias de entrada.

- ④ Se calculan los tiempos promedios y las desviaciones estándar.

## Resultados y discusión

En el Cuadro 4 se presentan los ECM y EAM de los siete filtros digitales para audio implementados. En el Cuadro 5 se muestran los tiempos promedios consumidos en el procesamiento de las muestras de audio y su respectiva desviación estándar. Finalmente, en la Figura 4 se muestran las respuestas en frecuencia de los filtros.

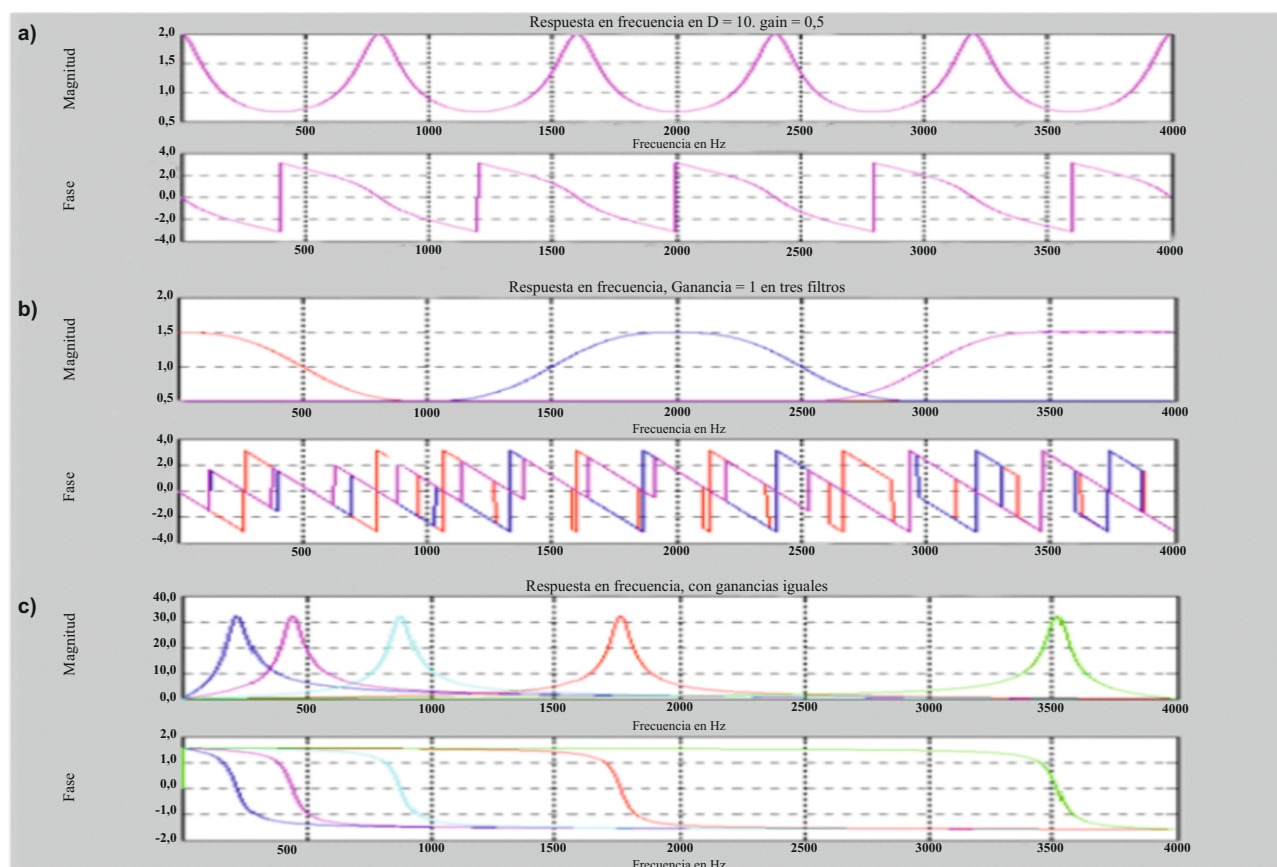
Del Cuadro 4, se puede observar que el ECM y EAM son valores pequeños para los siete filtros implementados, indicando con esto que la diferencia en promedio es pequeña entre los valores obtenidos por la simulación implementada en Matlab® y los resultados obtenidos del DSP.

**Cuadro 4.** Resultados ECM y EAM

Efecto de Audio	ECM	EAM
Reverberación Estudio	0,0018	0,022
Reverberación Auditorio	0,0015	0,031
Reverberación Sala de concierto	0,0021	0,032
Reverberación Iglesia	0,0025	0,037
Ecualizador Tres Bandas	0,0012	0,019
Ecualizador Gráfico	0,0013	0,020

En el Cuadro 5 se puede resaltar que el procesamiento no presenta retardos, lo cual quiere decir que a cada muestra de audio se ejecuta en menos de 125  $\mu\text{seg}$  y teniendo en cuenta que la frecuencia de muestreo es de 8 KHz se puede decir que el procesamiento se realiza en tiempo real.

De la figura de respuesta en frecuencia de los filtros se puede notar que se obtiene el diseño de acuerdo a las frecuencias que se desean



**Figura 4.** a) Respuesta en frecuencia del filtro para la reverberación, b), Respuesta en frecuencia de los tres filtros del ecualizador con ganancias iguales, c) Respuesta en frecuencia de los filtros del ecualizador gráfico con ganancias iguales.

modificar, por lo tanto se pueden lograr en un 100% los efectos propuestos.

**Cuadro 5.** Resultados de Tiempo de Procesamiento de las muestras de audio.

Efecto de Audio	Tiempo Promedio ( $\mu$ s)	Desviación Estándar ( $\mu$ s)
Reverberación Estudio	80,2	1,2
Reverberación Auditorio	85,5	1,5
Reverberación Sala de concierto	90,1	1,1
Reverberación Iglesia	92,4	0,9
Ecuilizador Tres Bandas	78,8	1,4
Ecuilizador Gráfico	79,5	1,6

De igual manera, se validaron los efectos de audio sometiendo a un oído humano no entrenado para que percibiera los efectos implementados y comentara si se lograba el efecto. Para todos los filtros digitales se logró que el oído no entrenado notara el procesamiento propuesto, logrando de esta manera la implementación de un procesador para efectos de audio en tiempo real.

Se implementaron los efectos de audio: reverberación y ecualización. Para la reverberación se simulaban cuatro ambientes y para la ecualización se implementó un ecualizador de tres bandas y un ecualizador gráfico de cinco bandas. Todos los efectos son notorios para un oído humano no entrenado y los filtros digitales se pudieron implementar en tiempo real considerando que la frecuencia de muestreo empleada en el ADC fue de 8 KHz.

Los sistemas basados en procesadores, por más optimizada que sea la arquitectura interna, tiene limitaciones en tiempo de procesamiento y memoria. Por lo tanto, es muy importante optimizar el código para la implementación de cualquier sistema en tiempo discreto.

Finalmente, se recomienda se exploren en trabajos futuros el procesamiento de señales tales como: imágenes, videos y datos.

## Literatura citada

Ackenhusen, J. 1999. Real-Time Signal Processing:

The design and implementation of signal processing systems. Prentice Hall, New York. pp. 20 - 80.

Barrero, F; Ruiz, F & Toral, S. 2005. Procesadores Digitales de Señal de Altas Prestaciones de Texas Instruments. McGraw-Hill. 520 p.

Chassaing, R. 2002. DSP applications using C and the TMS320C6X DSK. John Wiley & Sons, INC. 360 p.

Embree, P. 1995. C algorithms for real - time DSP. Prentice Hall. 237 p.

Gottfried, B. 1998. Programación en C. Segunda edición. Mc Graw Hill. 670 p.

Grijota, J. 2008. Implementación de un Procesador Digital de Efectos Mediante DSP e Interfaz Gráfica sobre Plataforma W2000 Y XP. UPC. EUETIT. TERRASSA. pp. 14 - 30.

Landa, O; Loyo, J & Rodríguez, J. 2007. Aplicación de Audio Musical Utilizando un DSP. México D.F. pp. 1 -28.

Marín, J. 2002. Procesamiento Digital de Señales. Universidad del Quindío. pp. 7 - 83.

Oppenheim, A & Sillsky, A. 1998. Señales y Sistemas. Segunda Edición. Prentice Hall. 992 p.

Oppenheim, A & Schafer, R. 2000. Tratamiento de Señales en Tiempo Discreto. Segunda edición. Prentice Hall Iberia. 873 p.

Pérez, A. 2006. Estudio de Efectos de Audio para Guitarra, e Implantación Mediante DSP. Escuela Técnica Superior de Ingeniería. pp. 4 - 83.

Proakis, J & Manolakis, D. 1998. Tratamiento digital de señales. Tercera edición, Prentice Hall, Madrid. pp. 509 - 784.

Sidney, C; McClellan, J; Oppenheim, A; Parks, T; Schafer, R & Schuessler, H. 1998. Ejercicios de tratamiento de la señal utilizando Matlab V.4. Prentice Hall. pp. 45 - 368.

Smith, S. 1997. The Scientist and Engineer's Guide to Digital Signal Processing. San Diego California. pp. 1 - 20.



Texas Instruments. 2000. Datasheet  
TMS320c6711, SPRS073D. 132p.

Wanhammar, L. 1999. DSP Integrated circuits.  
Academic Press. pp. 59 - 383.

---

**Luis Gabriel Marín Collazos**

Ingeniero Electrónico de la Universidad Autónoma de Colombia. Especialista en Evaluación Pedagógica de la Universidad Católica de Manizales. Candidato a Magister en Ciencias de la Información y las Comunicaciones de la Universidad Distrital Francisco José de Caldas. Docente ocasional tiempo completo de la Universidad de la Amazonia. Su área de trabajo son los sistemas de control y la inteligencia computacional. Pertenece al grupo de investigación GIECOM de la Universidad de la Amazonia.

**Autor para correspondencia**

**E-mail: lgmarin@uniamazonia.edu.co.**